Google Advanced Search Preferences Language Tools Search Tips

printf trace buffer Google Search

Web · Images · Groups · Directory · News-New! ·

Searched the web for **printf trace buffer**. Results **1** - **10** of about **21,000**. Search took **0.05** seconds.

Interdata_v6/usr/sys/**trace**.s

... extrn trmask,trbuff extrn **printf** * * **trace**(mask, label, value) * char *label; * * - log message label (first 4 chars) & value in * circular **trace buffer** "trbuff ...

minnie.tuhs.org/UnixTree/Interdata_v6/ usr/sys/trace.s.html - 3k - Cached - Similar pages

MPI Programming Guide

... **printf**("Could not stop tracing!\n"); } / Flush the memory **buffer** to disk */ if(0!=VT_trc_flush_c()) { **printf**("Could not flush **trace buffer**!\n"); } for(i=0;i ...

hpc.snu.ac.kr/document/aixdocumentlibrary/en_US/ a_doc_lib/sp32/pe/html/d3d70mst41.html - 10k - Cached - Similar pages

GDB Introspect **Trace** Debugging: Commands

... record called a **trace** frame in a **buffer** called the **trace buffer**. ... you can use commands such as GDB's **printf** command to format the collected **trace** data any ...

sources.redhat.com/gdb/talks/ esc-west-1999/commands.html - 19k - Cached - Similar pages

Usage Notes for **TRACE**

... **TRACE** provides similar functionality to **printf** or printk, but with significant additional ... addition to the user's string, each **trace** in the **trace buffer** is time ...

linux-rep.fnal.gov/new.trace.notes.html - 11k - Cached - Similar pages

Example Code

... SlkParse (options & DISPLAY, SlkActions, peeker, error, log, 0); **printf** ("Answer = %f ... n\n\t\tPARSE DERIVATION\n\n", 0); log.**trace** (**buffer**); } for (symbol ...

home.earthlink.net/~parsersinc/slk/example.html - 19k - Cached - Similar pages

www.geocities.com/lme3623/410/TTT/manager.txt

... GAME_OVER"); write(hold, **buffer**, strlen(**buffer**) + 1);}} while (1 ... map[x] == ' ' **trace** = 0;}} if (**trace** == 2){ return ... 0);} void Output(void){ **printf**("\n"); **printf** ...

4k - Cached - Similar pages

home.planet.nl/~faase009/Ha_bf_c.txt

... s}\n", depth, depth, "") ; break ; default: **printf**("/*ERROR*/\n ... int argc, char *argv[]) { char **buffer**[10000] ; int ... argv[i], "-t") == 0) **trace** = 1 ; else if ...

11k - Cached - Similar pages

Midterm I - 1999

... char_ptr; struct node * new; fgets(**buffer**, 80, stdin); char_ptr = **buffer**; while (*char_ptr != '\n') { func(*char_ptr); char_ptr++; } **printf**("**Trace** the program ...

davinci.newcs.uwindsor.ca/F2002/60-212/ solutionExams/solmid1_99.htm - 17k - Cached - Similar pages

Program to read the **trace** files on the textbook CD

... unsigned char tracedata[MAXREC]; /* **buffer** to hold record data */. } tracerec;. ... **printf**("Enter the filename of the **trace** data >");. gets(filename);. ...

williams.cs.ncat.edu/Networks/readtrace.htm - 52k - Cached - Similar pages

CADRE **Trace** Libraries:Preprocessed Code Sample

... I/O calls with corresponding **trace** library versions. ... if (fp !=NULL) { do { cnt = fread(**buffer**, sizeof(char ... no write perm) */ fclose(fp); } **printf**("Read %d ...

www-pablo.cs.uiuc.edu/Project/CADRE/CADRETraceLibraries/ samplecode/UNIXIOPreprocessedInstrumentation.htm - 15k - Cached - Similar pages

Searched Groups for **"trace macro"**.

Results **1 - 10** of about **614**. Search took **2.51** seconds.
Sorted by relevance Sort by date

Re: **TRACE macro**
/ Is it possible to write a **trace macro** that, when included inside a function, / prints some string and where it was called from? void tracePosition(char *file ...
comp.lang.c - Dec. 12, 1999 by Once in a China Blue Moon - View Thread (6 articles)

MCL & the **trace macro**
... a function only when its second argument is zero. I read a bit more about the **trace macro** in the MCL Reference (p. 360). I found an example on p. 362 which ...
comp.lang.lisp.mcl - Jun. 2, 1995 by Brian Forney - View Thread (1 article)

TRACE macro
As near as I can tell, the undocumented **TRACE macro** is broken. If you ever pass it a string with over about 200 bytes (the number appears to vary depending on ...
borland.public.cppbuilder.language - Jan. 28, 1998 by Howard Lee Harkness - View Thread (3 articles)

How to write a Ex-**TRACE Macro**?
... while debuging multi-thread program. it seems the Stand **TRACE Macro** can't do so, as a result i write a Ex-**TRACE macro** to finish my work. i declare some local ...
microsoft.public.vc.debugger - Feb. 16, 2002 by SmokyRain - View Thread (4 articles)

Re: Why is MFC **TRACE macro** losing characters?
... do to make this buffer larger. The only real workaround is to modify **TRACE macro** so that it prints to a file instead of to OutputDebugString. Then you can use ...
microsoft.public.vc.language - Sep. 24, 2001 by Hans Dietrich - View Thread (7 articles)

Problem with the **TRACE macro** in checks.h ?
When I call the **TRACE**("my messge here") **macro** that's in checks.hI've found that successively smaller messages contain trailing bits of text from preceeding ...
comp.os.ms-windows.programmer.tools.owl - May. 17, 1996 by John Collins - View Thread (2 articles)

PRB: MSVC++ 1.5 **TRACE Macro**
Problem with Microsoft VC++ v 1.5 (MFC v2.0) Professional **TRACE macro** Please can somebody help? I've just spent 6 days trying to debug one of my programs and ...
comp.os.ms-windows.programmer.tools.mfc - Dec. 15, 1996 by Simon Hughes - View Thread (2 articles)

Re: Q: **TRACE macro** in Release
... would work correctly in Debug mode) It's not that simple. If I substitute "" for the **TRACE macro** above, I get: if (x == 0) ; SomeFunc(x); No you get if(x == 0 ...
microsoft.public.vc.language - May. 29, 2001 by Ron Natalie - View Thread (13 articles)

Re: **TRACE macro** problem
... always TRUE' I didn't have that problem, but I did notice that in BCB1, the **TRACE macro** was broken, so I wrote my own. The problem was that if you passed **TRACE** ...

Search Results

Search Results for: **[printf <and> "trace buffer"]**
Found **4** of **105,146 searched.** > Rerun within the Portal

Search within Results

> Advanced Search > Search Help/Tips

Sort by: **Title** **Publication** **Publication Date** **Score** Binder

Results 1 - 4 of 4 **short listing**

1 Parasight: a high-level debugger/profiler architecture for shared-memory multiprocessor 80%

Z. Aral , Ilya Gertner

Proceedings of the 2nd international conference on Supercomputing June 1988

Existing debuggers and profilers are inadequate for debugging and profiling parallel programs. They are awkward in their handling of multiple threads of control and highly intrusive in their monitoring of program behavior. ParasightTM is an architecture that is geared towards non-intrusive high-level debugging and profiling. Parasight controls and observes the execution of parallel programs in terms of the set of abstractions that are being employed by the programmer. D ...

2 High-level debugging in parasight 77%

Ziya Aral , Ilya Gertner

ACM SIGPLAN Notices , Proceedings of the 1988 ACM SIGPLAN and SIGOPS workshop on Parallel and distributed debugging November 1988

Volume 24 Issue 1

Debugging parallel programs with time critical dependencies is difficult due to subtle race conditions that may cause deadlock, starvation, and other errors. These errors can be detected by multiple instrumentation points triggered by logical assertions. Although some advanced debuggers provide a programmer with the ability to define complex logical assertions, they are inadequate for debugging parallel programs due to the high overhead of monitoring these assertions. This paper ...

3 From trace generation to visualization: a performance framework for distributed parallel systems 77%

C. Eric Wu , Anthony Bolmarcich , Marc Snir , David Wootton , Farid Parpia , Anthony Chan , Ewing Lusk , William Gropp

Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM) November 2000

In this paper we describe a trace analysis framework, from trace generation to visualization. It includes a unified tracing facility on IBM SP systems, a self-defining interval file format, an API for framework extensions, utilities for merging and statistics generation, and a visualization tool with preview and multiple time-space diagrams. The trace environment is extremely scalable, and combines MPI events with system activities in the same set of trace files, one for each SMP node. Sin ...

4 Improving instruction supply efficiency in superscalar architectures using instruction 77%

From: Maksim P (maksimpogorelov@practiceworks.com)
Subject: Re: How to redirect output?
Newsgroups: microsoft.public.vc.language
Date: 2002-04-30 06:53:58 PST

Search Result 7

View: Complete Thread (4 articles)
Original Format

```
If you need it for debugging purposes,
TRACE() macro can be redirected to a file in the debug mode using
_CrtSetReportMode().

"Edward" <konvalo@263.net> wrote in message
news:#9APoeE8BHA.2052@tkmsftngp04...
> Hello,
>   I want to redirect output of printf(...) and perror(...) to a file,for
> example:
>
> printf("%s",buffer);
> perror("Error when receiving...");
>
> I want to write the display the result of printf and perror function to a
> disk file,such as " log.txt". When I run any printf and perror statement in
> my program,it also write the output to a file.How to do it? Any ideas will
> be appreciated!
> Thanks in advance.
> Regards,
> Edward
>
>
>
>
```

Groups search result 3 for **"trace macro" printf**

From: JDD (jeff.delaney@eastmansoftware.com) Search Result 3
Subject: Re: Debugging MTS-components
Newsgroups: microsoft.public.microsofttransactionserver.programming View: Complete Thread (12 articles)
Date: 1998/07/28 Original Format

Here is text from the Windows NT 4.0 Option Pack on MTS Debugging

Debugging Visual C++ MTS Components
You can use Visual StudioÔ 97 to debug MTS components written in Visual C++Ò, including components that call SQL Server functions or stored procedures. For more information, see Debugging Visual Basic MTS Components.

The following information applies to components that have their activation property set to In a dedicated server process.

Microsoft Transaction Server supports the COM transparent remote debugging infrastructure. If transparent remote debugging is enabled, then stepping into a client process will automatically stop at the actual object's code in the server process, even if the server is on a different computer on the network. A debugging session is automatically started on the server process if necessary. Similarly, single stepping past the return address of code in a server object will automatically stop just past the corresponding call site in the client's process.

In Microsoft Visual C++, selecting the OLE RPC debugging check box (on the Tools menu, select the Options submenu and choose the Debug property sheet) enables transparent remote debugging. It is not known at this time whether other debuggers support this infrastructure.

You can also debug your Microsoft Transaction Server component DLL in Visual C++ by performing the following steps. Each of these steps is made either inside the MTS Explorer or inside of a Visual C++ session with your MTS DLL project.

Shutdown server processes using the MTS Explorer. To do this, right-click My Computer, and select Shutdown Server Process.

In your Visual C++ session, under Project, Settings, Debug, General, set the program arguments to the following string: "/p: PackageName", for example: /p: "Sample Bank"

In the same property sheet, set the executable to the full path of the Mtx.exe process, for example: "c:\MTx\MTx.exe".

Set breakpoints in your component DLL, and you are ready to debug.

Run the server process (in the Build menu, select Start Debug and click Go.)
The following information applies to in-process component DLLs that have their activation property set to In the creator's process.

You can debug your in-process MTS component DLL in Visual C++ by performing

the following steps. Each of these steps is made inside a Visual C++ session with your base process project.

Set the component DLL under Build, Settings, Debug, Additional DLLs.

Now you are ready to step into or set breakpoints in your component DLL at will.
If you are using Visual Studio and Microsoft Foundation Classes (MFC) to debug, the **TRACE macro** can facilitate your debugging. The **TRACE macro** is an output debug function that traces debugging output to evaluate argument validity. The **TRACE macro** expressions specify a variable number of arguments that are used in exactly the same way that a variable number of arguments are used in the run-time function **printf**. The **TRACE macro** provides similar functionality to the **printf** function by sending a formatted string to a dump device such as a file or debug monitor. Like **printf** for C programs under MS-DOS, the **TRACE macro** is a convenient way to track the value of variables as your program executes. In the Debug environment, the **TRACE macro** output goes to afxDump. In the Release environment, the **TRACE macro** output does nothing.

Example:

```
// example for TRACE
int i = 1;
char sz[] = "one";
TRACE( "Integer = %d, String = %s\n", i, sz );
// Output: 'Integer = 1, String = one'
```

The **TRACE macro** is available only in the debug version of MFC, but a similar function could be written for use without MFC. For more information on using the **TRACE macro**, see the "MFC Debugging Support" section in Microsoft Visual C++ Programmer's Guide.

Note that you should avoid using standard ASSERT code in Visual C++.
Instead, it is recommended that you write assert macros like a MessageBox using the MB_SERVICE_NOTIFICATION flag, and **TRACE macro** statements using the OutputDebugString function call.

Henrik Knutsson wrote in message <6pk663$78o@newstoo.ericsson.se>...
>Hi,
>
>I'm developing MTS-components from visual C++. As long as I'm debugging my
>components from a client outside the MTS as normal CLSCTX_INPROC_SERVER
>DLL's there's no problem. But when I place the components in the MTS and
>wants to debug them from a client, so I can debug the specific MTS
>functuality i.e SafeRef(), GetObjectContext etc, I always steps into the
>asssambler-code when i steps into the component from the breakpoint in the
>client-code. How can I debug SafeRef(), GetObjectContext etc??
>Is it possible at all to debug components if they run in the MTS??
>
>Please help me.
>
>Sincerly
>Henrik Knutsson
>
>